SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered) UNCLASSIFIED

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | | |
|---|---|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO.<br>AD-A088 767 | 3. RECIPIENT'S CATALOG NUMBER |

4. TITLE (and Subtitle)
(6) FLOW CONTROL AND ROUTING ALGORITHMS FOR DATA NETWORKS.

5. TYPE OF REPORT & PERIOD COVERED
(9) Paper-Technical papers,

6. PERFORMING ORG. REPORT NUMBER
(14) LIDS-P-1013

7. AUTHOR(s)
R. G./Gallager ~~and~~ S. J./Golestaani (10)

8. CONTRACT OR GRANT NUMBER(s)
(15) ARPA Order No. 345/5-7-75
ONR/N00014-75-C-1183,
ARPA Order-345

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Massachusetts Institute of Technology
Laboratory for Information and Decision Systems
Cambridge, Massachusetts 02139 410950

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Program Code No. 5T10
ONR Identifying No. 049-383

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

12. REPORT DATE
(11) July 1980

13. NUMBER OF PAGES
6 (12) 8

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
Office of Naval Research
Information Systems Program
Code 437
Arlington, Virginia 22217

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Aproved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We consider flow control algorithms consisting of two parts: quasi-static flow control and dynamic flow control. The quasi-static part uses short term average information on network utilization to allocate maximum data rates and to determine routes for each user. The rates are allocated to achieve an optimal trade-off between assigned priority cost functions for each user and the cost of congestion in the network. This optimization can be done by a distributed algorithm and is essentially no more complicated than optimizing routing alone. The dynamic flow control has the function of admitting or rejecting individual units of traffic

next page

410950

20. (Continued)

into the network so as to enforce the maximum allocated rates and to prevent congestion by smoothing out the fluctuations in buffer occupancy.

# FLOW CONTROL AND ROUTING ALGORITHMS FOR DATA NETWORKS

R. G. Gallager* and S. J. Golestaani**

*Mass. Inst. of Tech., Cambridge, Ma. 02139; **Isfahan Un. of Tech., Isfahan, Iran

July 1980

LIDS-P-1013

To be presented at ICCC'80, Atlanta Georgia, October 27-30, 1980.

## ABSTRACT

We consider flow control algorithms consisting of two parts: quasi-static flow control and dynamic flow control. The quasi-static part uses short term average information on network utilization to allocate maximum data rates and to determine routes for each user. The rates are allocated to achieve an optimal trade-off between assigned priority cost functions for each user and the cost of congestion in the network. This optimization can be done by a distributed algorithm and is essentially no more complicated than optimizing routing alone. The dynamic flow control has the function of admitting or rejecting individual units of traffic into the network so as to enforce the maximum allocated rates and to prevent congestion by smoothing out the fluctuations in buffer occupancy.

## I. INTRODUCTION

The sources of a data network can be qualitatively modeled as random processes with slowly varying statistics. Thus, for example, on a time frame of seconds and milliseconds, we would focus on individual message arrivals into the network, whereas on a time frame of minutes and seconds, we would focus on the changing message arrival rates and the initiations and terminations of user sessions. If the network is moderately to heavily loaded, then both random fluctuations and the changing statistics can cause the inputs to temporarily overload the message handling ability of the network. If the network has no control over the inputs, then the overloads will cause the network buffers to fill up; this causes inefficient utilization of the communication lines, the throughput drops rapidly, and the network is congested; even if the input rapidly drops back to normal levels, the network remains congested.

It should be clear from the above that a network with finite buffers and communication capacity requires some means of controlling the inputs (if only to throw away messages that cannot be buffered). Flow control, as used here, is the set of techniques used by the network to control the inputs. An excellent review and bibliography of flow control techniques is given by Gerla and Kleinrock [1]. Our treatment here will ignore what [1] refers to as transport level flow control (i.e., control between user processes), since that is a separable issue. We also ignore buffer management strategies except to the extent that they exert control on the inputs. Finally we do not distinguish between internal congestion in the network and congestion caused by a destination that is slow in absorbing messages from the network. This latter form of congestion can be viewed as internal congestion by modelling the path from destination node to external destination as a low capacity network link.

The general approach of flow control strategies is quite clear; one wants to curtail the inputs when the buffers in the network are filling up. There are two difficulties in this apparently simple approach. The first is that instantaneous global knowledge about the network is not available at the nodes; providing even partial information about the state of the network requires resources otherwise available for increasing throughput. The second is the issue of fairness and priorities; if inputs must be curtailed, which inputs should be curtailed and by how much?

# FLOW CONTROL AND ROUTING ALGORITHMS FOR DATA NETWORKS

R. G. Gallager* and S. J. Golestaani**

*Mass. Inst. of Technology, Cambridge, Ma. 02139; **Isfahan Un. of Tech., Isfahan, Iran

We consider separating flow control into a quasi-static and a dynamic part. The quasi-static part determines allowable transmission rates for each session, considering priorities, fairness, and the expected level of congestion throughout the network. The dynamic part has the function of admitting or rejecting individual messages to the network so as to satisfy the above allowable rates and to smooth out the fluctuations in arrivals and buffer occupancies. Conventional flow control strategies are primarily dynamic according to this division, and their quasi-static part is generally limited to rejecting new sessions when the system is heavily loaded.

As an example, consider a network with end to end windowing where each session has a window of w packets. If there are m sessions each transmitting packets as fast as possible, then on the order of mw packets will be buffered in the network (temporarily ignoring acknowledgement times). As m increases, the network will eventually become congested. Conventional strategies can sometimes avoid this congestion by rejecting new sessions, but if m increases due to inactive sessions becoming more active, the buffers will eventually fill up. The problem is that as the loading increases, delay increases, and the rate of each session is decreased, but not quite enough to prevent congestion. What is needed in such a situation is a means of adapting the window sizes to varying network conditions. We show in section 3 how to use the quasi-static flow control developed in section 2 to adapt window sizes to optimal values (according to a flexible criterion of optimality).

## II. QUASI-STATIC FLOW CONTROL AND ROUTING

Our objective in this section is to determine an optimal set of input rates for a network of given topology with given user demands. We take the viewpoint of a supplier of network resources who can lose revenue either by failing to meet user demands or by causing long delays and lost messages due to congestion. Thus for each session using the network, we create a cost function, e(r), which is a decreasing function of the rate r allocated to the given session (see Figure 1). As r is decreased, the cost in user dissatisfaction clearly increases. We discuss the implications of different forms of this function later, but for the time being simply restrict it to be twice differentiable and convex (i.e., the second derivative is non-negative).

Similarly, for each communication link of the network, we create a cost function g(f), (see Figure 2) which is an increasing function of the traffic flow F on that link. As the flow approaches the capacity of the link the expected queue length of messages waiting to traverse the link increases and the danger of congestion increases; g(F) should represent the cost of this congestion danger. It seems appropriate to choose g(F) to approach ∞ at the maximum link flow that can be handled, given the number of buffers in the node. Typically this maximum F will be somewhat smaller than the capacity of the link, and we can call it the effective capacity, $C_e$, of the link.

A reasonable choice for g(F), then, in keeping with the Kleinrock independence assumption for network queueing, is

$$g(F) = \frac{F}{C_e - F} \qquad (1)$$

Our objective, now, is to form an aggregate cost function as the sum of the individual session cost functions and link cost functions. We then choose the session rates and network routes to minimize the aggregate cost function. In the remainder of this section, we set up the minimization problem precisely, give its solution, and then discuss the choice of session cost functions. Section three shows how the optimization can be implemented in a distributed way and relates the quasi-static solution to dynamic algorithms.

Suppose that the network has N nodes (switches) denoted by the integers 1,...,N, and for notational simplicity, assume that for each pair (i,j) of nodes, there is at most one user session originating at node i and destined for node j. Let $r_{ij}^d$ be the rate (in bits per second) desired by the user for this session and let $r_{ij}$ be the rate allocated by the network. Thus

$$0 \leq r_{ij} \leq r_{ij}^d \qquad (2)$$

If there is no session from i to j, or if the session is currently inactive, we take $r_{ij}^d = 0$.

Let the network have L directed links denoted 1,...,L. Each link goes out from some node i into some node j. Let O(i) be the set of links going out from node i, and let I(i) be the set of links going into node i. A full duplex communication channel between nodes i and j is regarded as two links, one going out from i and into j and the other going out from j and into i.

Let $F_\ell$ be the user traffic flow (in bits/sec.) on link $\ell$ and let $f_{\ell j}$ be the part of that traffic belonging to sessions with destination node j. We then have the constraint equations:

$$\sum_{j=1}^{N} f_{\ell j} = F_{\ell} \; ; \; 1 \le \ell \le L \qquad (3)$$

$$f_{\ell j} \ge 0 \; ; \; 1 \le \ell \le L, \; 1 \le j \le N \qquad (4)$$

$$r_{ij} + \sum_{\ell \epsilon I(i)} f_{\ell j} = \sum_{\ell \epsilon O(i)} f_{\ell j} \; ; \quad 1 \le i,j \le N; \; i \ne j \qquad (5)$$

Equation (3) states that the total user traffic is simply the sum of the traffics to all of the destinations. Equation (5) states that the total traffic coming into node i destined for node j (the exogenous traffic $r_{ij}$, plus the traffic passing through i from other nodes) is equal to the total traffic going out. If $f_{\ell j}$ was taken to be a short term average, then (5) would not be quite correct, and in fact the left side minus the right side would represent the rate of buffer buildup at node i for traffic going to j. Our viewpoint here is somewhat different, and is in fact the essence of the quasi-static approach. We are allocating session rates, and choosing flows which would be feasible over the long term with stationary inputs at the allocated rates. Any set of flows satisfying (3)-(5) can be achieved for the given input rates by a routing policy which, for each node i and destination j, routes the traffic to j out of i on outgoing links proportionally to the terms on the right hand side of (5); conversely the long term flows resulting from any routing policy (given that data is not thrown away inside the network) will satisfy (3)-(5).

The objective function that we want to minimize is given by

$$J(\vec{F},\vec{r}) = \sum_{\ell=1}^{L} g_{\ell}(F_{\ell}) + \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j \ne i}}^{N} e_{ij}(r_{ij}) \qquad (6)$$

where $g_{\ell}(F_{\ell})$ is the cost of congestion with flow $F_{\ell}$ on link $\ell$ and $e_{ij}(r_{ij})$ is the cost of restricting the session i,j to the rate $r_{ij}$. Analytically, our problem is to minimize $J(\vec{F},\vec{r})$ subject to the linear constraints (2)-(5). Applying Kuhn-Tucker theory, we are led to the following necessary and sufficient conditions for minimizing J. [2,3].

<u>Theorem 1</u>: Assume that $g_{\ell}$ and $e_{ij}$ have first and second derivatives satisfying $g'_{\ell} > 0$, $g''_{\ell} > 0$, $e'_{ij} < 0$, $e''_{ij} > 0$ for $1 \le \ell \le L$, $1 \le i,j \le N$, $i \ne j$. Necessary and sufficient conditions on $\vec{f}^*$, $\vec{F}^*$, $\vec{r}^*$ to minimize (6) subject to (2)-(5) are that a set of numbers $\lambda_{ij}$, $1 \le i,j \le N$ exist with $\lambda_{ii} = 0$, $1 \le i \le N$, such that a) for all i,j,k,

$$g'_{\ell}(F^*_{\ell}) + \lambda_{kj} \ge \lambda_{ij} \qquad (7)$$

for $\ell \epsilon I(k)$, $\ell \epsilon O(i)$, with equality if $f^*_{\ell j} > 0$,

and b), for all i,j such that $r^d_{ij} > 0$,

$$e'_{ij}(r^*_{ij}) + \lambda_{ij} \begin{cases} = 0 & \text{for } 0 < r^*_{ij} < r^d_{ij} \\ \le 0 & \text{for } r^*_{ij} = r^d_{ij} \\ \ge 0 & \text{for } r^*_{ij} = 0 \end{cases} \qquad (8)$$

If multiple sessions go from i to j, then $r_{ij}$ in constraint (5) should be replaced by the sum of the allocations for i,j sessions, and (8) must be satisfied individually for each session.

We can interpret $g'_{\ell}(F^*_{\ell})$ as the incremental cost of traffic on link $\ell$. If there is a path carrying traffic with destination j from node i to node j, then $\lambda_{ij}$ is the incremental cost of traffic on that path. Equation (7) then states that all traffic flows on paths of minimum incremental cost. If one views $g'_{\ell}(F^*_{\ell})$ as the "length" of link $\ell$, then (7) states that all traffic flows by shortest routes. This condition is well known when the rates are fixed and one is optimizing only over routing.

We will call $-e'_{ij}(r_{ij})$ the <u>priority function</u> for session i,j. It is the incremental gain for additional allocation to i,j. Equation (8) states that $r^*_{ij}$ is set between 0 and $r^d_{ij}$ to make this incremental gain as close as possible to $\lambda_{ij}$ (i.e., equal to $\lambda_{ij}$ except at the end points).

Note that the optimality conditions (7) and (8) depend only on the incremental link costs $g'_{\ell}$ and the priority functions $-e'_{ij}$. This means that arbitrary constants could be added to $g_{\ell}$ or $e_{ij}$ without affecting the optimum. Note also that the optimum point is independent of $r^d_{ij}$ over the range of $r^d_{ij} > r_{ij}$. This is a desirable feature for a flow control strategy, preventing users who are being actively flow controlled from attempting to increase their share of the resources by increasing their demands.

The maximum average network loading permitted by the flow control can be adjusted by scaling the priority functions. This scaling factor should depend, of course, on how effective the dynamic component of the flow control is in preventing congestion for a given short term average load level.

Some insight into the choice of priority functions can be obtained by considering the priorities of the form

$$-e'_{ij}(r_{ij}) = \left(\frac{a_{ij}}{r_{ij}}\right)^{b_{ij}} \qquad (9)$$

The factor $a_{ij}$ in (9) should be taken as a measure of the typical rate required for the session. For example, two sessions between nodes i and j with a given value of $a_{ij}$ are equivalent to a single session with twice the above value for $a_{ij}$. The factor $b_{ij}$ is more a measure of the importance of the session. The larger $b_{ij}$ is, the less the session will be cut back with increasing network loading. Naturally, if $b_{ij}$ is large for all sessions, this effect disappears, since the incremental costs simply increase with increasing demands until the allocated rates are feasible.

In considering algorithms to solve for the optimum rates and routes of (7) and (8), we start with a very general approach that reduces the problem to a quasi-static routing problem. For eacn session i,j, consider a fictitious communication link ℓ' that carries all of the traffic, $r^d_{ij} - r_{ij}$, that is offered by the user but rejected by the flow control. Letting O'(i) be the outgoing links from node i including these fictitious links, the flow constraint equation (5) becomes:

$$r^d_{ij} + \sum_{\ell\epsilon I(i)} f_{\ell j} = \sum_{\ell\epsilon O'(i)} f_{\ell j} \qquad (5')$$

Now consider imposing a cost function on the fictitious link ℓ' that carries the session i,j rejected traffic (see fig. 3):

$$g_{\ell'}(F_{\ell'}) = e_{ij}(r^d_{ij} - F_{\ell'}) \qquad (10)$$

Since $r^d_{ij} - F_{\ell'}$ is just the accepted traffic for session i,j, $g_{\ell'}(F_{\ell'})$ is the same as $e_{ij}(r_{ij})$. This means that the cost function (7) is equal to $\sum_\ell g_\ell(F_\ell)$ when the sum is taken over real and fictitious links.

The minimization of this cost function subject to (2)-(5') is now just a conventional quasi-static routing problem. In simple terms, we have replaced the question of how much traffic to allocate to session i,j with the equivalent question of how to apportion the traffic between the real and fictitious links. There are a variety of distributed algorithms for optimizing quasi-static routing in the literature [4]-[7], and any of them can now be used to jointly optimize quasi-static flow control and routing.

## III. DYNAMIC QUASI-STATIC FLOW CONTROL

The major difficulty with considering flow control in terms of fictitious links in a routing problem is that this approach is not easily combined with dynamic flow control. To illustrate both this problem and possible alternative approaches, consider the use of end to end windowing for dynamic flow control [1], [8]. In this situation, the allocated traffic rates are determined by the window sizes. Thus the quasi-static part of the flow control is to choose the window sizes that lead to the optimum rate allocation. Unfortunately, if a single window size is changed in a network, this typically changes delays throughout the network, and thus changes the allocated rates for all the sessions.

We can find the change in rates for a given change in window sizes by ignoring flow control for the moment and finding the relation between the expected number of packets outstanding in the network for each session and the rate of each session, assuming constant routes. Number the sessions from 1 to m, letting $r_m$ be the rate of session m and $q_{m\ell}$ be the fraction of session m to use link ℓ (note that for unifilar routes, $q_{m\ell}$ takes on only 0 or 1 values). Let $t_\ell(F_\ell)$ be the average delay per packet on link ℓ, including processing and queueing delays at the input to link ℓ. Let Γ be the average packet length (where by packet we mean the transmission unit in terms of which windows are defined). Then the average number of packets for session m at link ℓ is given by $r_m q_{m\ell} t_\ell(F_\ell)/\Gamma$. This quantity, summed over ℓ, is the average number of packets traversing the network at a given instant. Let $\alpha_m$ be the average delay for acknowledgements for session m. We assume priority for acknowledgement traffic so that $\alpha_m$ is independent of network delays. The averaged number of unacknowledged packets of session m out in the network is then

$$w_m = \frac{r_m}{\Gamma} [ \sum_{\ell} q_{m\ell} t_{\ell}(F_{\ell}) + \alpha_m ] \tag{11}$$

Substituting $F_{\ell} = \sum_m r_m q_{m\ell}$ into (11: and differentiating, we get the matrix equation

$$P = T + RQMQ^T \tag{12}$$

where $P_{mk}$ is $\Gamma \partial w_m/\partial r_k$, T is a diagonal matrix with $T_{mm} = \sum_{\ell} q_{m\ell} t_{\ell}(F_{\ell}) + \alpha_m$, R is a diagonal matrix with $R_{mm} = r_m$, Q is a matrix with elements $q_{m\ell}$, and M is a diagonal matrix with $M_{\ell\ell} = \partial t_{\ell}(F_{\ell})/\partial F_{\ell}$.

Equation (11) relates window sizes for sessions to rates under the assumption that each user transmits as fast as possible subject to the window limitation. Golestaani [2] shows that the matrix P is invertible, which means that incremental changes in window sizes lead to uniquely defined changes in session rates. He shows, moreover, that (11) uniquely specifies the rates in terms of the window sizes for given increasing functions $t_{\ell}(F_{\ell})$, given routing Q, and given $\alpha_m$ and $\Gamma$. Finally, he demonstrates a counter example to this uniqueness if acknowledgements are not sent at high priority.

Now consider the objective function (6) expressed as a function of the session rates $\vec{r}$ and the routing Q, $J^*(\vec{r},Q)$. It is easy to calculate $\partial J^*/\partial r_m$ as part of distributed routing algorithms [4]; it is simply the incremental cost of traffic on the routes for the session minus the priority function for the session. It is reasonable in a joint flow control and routing algorithm to increase (decrease) the window size for session *m* when $\partial J^*/\partial r_m$ is negative (positive). Assume in fact that one changes the window size of each session by an incremental amount $\Delta w_m = -\eta r_m \partial J^*/\partial r_m$. To first order, the change in J* is then

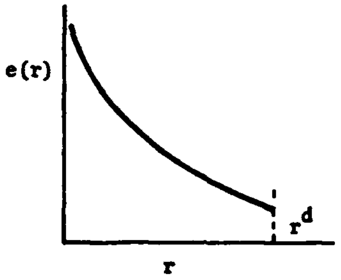
$$\Delta J^* = \sum_m - \frac{\partial J^*}{\partial w_m} \Delta w_m \tag{13}$$

Using (12), this becomes

$$\Delta J^* = - \frac{\eta}{\Gamma} (\nabla \cdot J^*)[TR + RQMQ^T R](\nabla \cdot J^*)^T$$

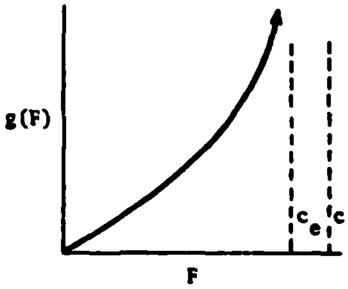
where $\nabla \cdot J^*$ is the gradient of J* with respect to w. Since T, M and R are positive diagonal matrices, it is easy to see that $\Delta J^*$ is negative if $J^* \neq 0$, and this approach leads to a descent algorithm.

In the above, we have shown that it is possible to vary window sizes in accordance with a distributed routing algorithm so as to reduce the quasi-static objective function. For simplicity, a number of details have been omitted, such as the maintenance of positive session rates, the need for an artiface to achieve incremental variations in integer window sizes, and the combined effect of routing and window variations. Our objective has not been to demonstrate a finished algorithm, but rather to point the way to a wide class of algorithms that combine a quasi-static method of allocating rates with a dynamic method of smoothing traffic.



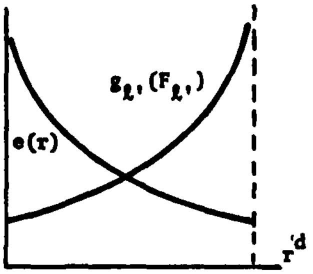
Cost e(r) of Rate Limitation r on a User

Figure 1



Congestion Cost of Link Flow F

Figure 2



Relation of Session Cost e(r) to Fictitious Link Cost
Figure 3

## REFERENCES

[1] M. Gerla and L. Kleinrock, "Flow Control: A Comparative Study", IEEE Trans. Commun., Vol. COM-28, pp. 553-574, April 1980.

[2] S. J. Golestaani, "A Unified Theory of Flow Control and Routing in Data Communication Networks", Report LIDS-TH-963, January 1980, Laboratory for Information and Decision Systems, Mass. Inst. of Tech., Cambridge, MA 02139.

[3] D. Luenberger, Introduction to Linear and Non Linear Programming, Addison-Wesley, 1973, Chapter 10.

[4] R. G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation", IEEE Trans. Comm., Vol. COM-25, pp. 73-85, January 1977.

[5] D. P. Bertsekas, "A Class of Optimal Routing Algorithms for Communication Networks", Proceedings of Fifth International Conference on Computer Communications (ICCC-80), November 1980.

[6] A. Segall, "Optimal Distributed Routing for Virtual Line-Switched Data Networks", IEEE Trans. Commun., Vol. COM-27, pp. 201-209, January 1979.

[7] T. E. Stern, "A Class of Decentralized Routing Algorithms Using Relaxation", IEEE Trans. Commun., Vol. COM-25, pp. 1092-1102, October 1977.

[8] R. Kahn and W. Crowther, "Flow Control in a Resource Sharing Computer Network", IEEE Trans. Commun., Vol. COM-20, pp 539-546, June 1972.

## ACKNOWLEDGEMENT

This research was conducted at the M.I.T., Laboratory for Information and Decision Systems with partial support provided by grant DARPA/ONR-N00014-75-C-1183 and grant NSF/ECS-7919880.

Robert G. Gallager received the S.B. degree in Electrical Engineering from the University of Pennsylvania in 1953 and the S.M. and Sc.D. degrees in Electrical Engineering from the Massachusetts Institute of Technology in 1957 and 1960 respectively. He has been at the Massachusetts Institute of Technology since 1956 and is currently Professor of Electrical Engineering and Computer Science and Associate Director of the Laborabory for Information and Decision Systems. He is also a consultant to Codex Corporation. Professor Gallager is the author of the text book Information Theory and Reliable Communication (Wiley and Sons, 1968), and was awarded the IEEE Baker Prize Paper Award in 1966 for the paper "A Simple Derivation of the Coding Thoerem and Some Applications". He is a member of the Board of Governors of the IEEE group on Information Theory, and was Chairman of the group in 1971. He is a Fellow of the IEEE and a member of the National Academy of Engineering. His major research interests are data communication networks, information theory, and communication engineering.

Seyyed J. Golestaani received the B.S. degree in Electrical Engineering from Arya-Mehr University of Technology in 1973 and the S.M. and Ph.D. degrees in Electrical Engineering and Computer Science from the Massachusetts Institute of Technology in 1976 and 1980 respectively. While studying at M.I.T., he was a research assistant in the Data Network Group, working in the areas of flow control, routing, and line protocols for error detection and retransmission. Since February 1980, he has been on the Electrical Engineering faculty at Isfahan University of Technology, Isfahan, Iran.

## Distribution List

Defense Documentation Center — 12 Copies
Cameron Station
Alexandria, Virginia 22314

Assistant Chief for Technology — 1 Copy
Office of Naval Research, Code 200
Arlington, Virginia 22217

Office of Naval Research — 2 Copies
Information Systems Program
Code 437
Arlington, Virginia 22217

Office of Naval Research — 1 Copy
Branch Office, Boston
495 Summer Street
Boston, Massachusetts 02210

Office of Naval Research — 1 Copy
Branch Office, Chicago
536 South Clark Street
Chicago, Illinois 60605

Office of Naval Research — 1 Copy
Branch Office, Pasadena
1030 East Greet Street
Pasadena, California 91106

New York Area Office (ONR) — 1 Copy
715 Broadway - 5th Floor
New York, New York 10003

Naval Research Laboratory — 6 Copies
Technical Information Division, Code 2627
Washington, D.C. 20375

Dr. A. L. Slafkosky — 1 Copy
Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, D.C. 20380

Office of Naval Research — 1 Copy
Code 455
Arlington, Virginia 22217

Office of Naval Research — 1 Copy
Code 458
Arlington, Virginia 22217

Naval Electronics Laboratory Center — 1 Copy
Advanced Software Technology Division
Code 5200
San Diego, California 92152

Mr. E. H. Gleissner — 1 Copy
Naval Ship Research & Development Center
Computation and Mathematics Department
Bethesda, Maryland 20084

Captain Grace M. Hopper — 1 Copy
NAICOM/MIS Planning Branch (OP-916D)
Office of Chief of Naval Operations
Washington, D.C. 20350

Advanced Research Projects Agency — 1 Copy
Information Processing Techniques
1400 Wilson Boulevard
Arlington, Virginia 22209

Dr. Stuart L. Brodsky — 1 Copy
Office of Naval Research
Code 432
Arlington, Virginia 22217

Captain Richard L. Martin, USN — 1 Copy
Commanding Officer
USS Francis Marion (LPA-249)
FPO New York 09501